

2020-06-17

Attendees

- Ram Jagadeesan
- Mic Bowman
- Hart Montgomery
- Clive Boulton
- Peter Somogyvari
- Luri Vinogradov

Notes

Self-describing characteristics for blockchain interchain integration and APIs.

The main discussion:

- Assumptions of trust from this Cactus diagram

<https://github.com/hyperledger/cactus/blob/master/whitepaper/use-case-ethereum-to-quorum-asset-transfer.png>

- Business model plugin handle

See last weeks IWG. See recording with @Peter and Vipin Bharathan <https://wiki.hyperledger.org/display/IWG/2020-06-10>

Personal feature branch where I've been dabbling with a plugin for Open ID Connect (still has to pass muster, but if you are interested)

<https://github.com/petermetz/blockchain-integration-framework/tree/feature/open-id-connect-web-service-plugin>

- Implicit assumptions
 - Identity
 - Government issue identity
 - KYC [Notable, State of Washington is currently dealing with so called 'Scattered Canary identity fraud discovered by security researchers <https://www.agari.com/email-security-blog/covid-19-unemployment-fraud-cares-act/>
 - Transfers
 - Trust Coinbase exchange
 - Real World Transfer
 - Scotch
 - Capture implicit outside of protocol
 - Code the smart contract based on half or transaction (trusting the scotch will arrive)
 - Requires external payment
 - R3
 - DoS
 - Identity agree to meta
 - Degrees of trust
 - All of nothing
 - \$5.00 is low risk
 - \$50 million is less absolute
 - Customise smart contract per transaction
 - Trust Cactus
 - Trust Cactus server hosting service
 - Host your own Cactus server
 - Cactus has ability to verify ledgers
 - Cactus has ability for plugins to verify each other
 - Cactus has ability to specify plugin used in smart contract? (Peter mention, unsure about this comment).

---//---

Future discussion (proposed agenda for later date)

- Trust NPM plugins?
- Mitigating Spectre v1 attacks is mainly needed for virtualized servers
- Isolating plugins with ES (observable-membrane)
- Isolating plugins with SES (agoric)

17 Jun 2020 Notes are fairly complete, but may have lost notion of context in brevity - cb